

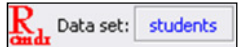
Praktikum 4

***R* and package *Rcmdr*: frequency tables, χ^2 - and Fisher exact test; correlation and regression analyses**

1.

1.1. Open the *R*, and then load the *R Workspace* (.Rdata-file) saved in last practical (*Fail -> Load Workspace ...*). If you haven't what to load, follow the points 1.2 and 1.3 or 1.4.

1.2. Open the *R Commander* (for example running the command `library(Rcmdr)`).

If you had the *R Workspace*, then it contains also the students dataset and in *R Commander* you should fix it as the default dataset for *R Commander* menus: .

1.3. If you haven't the *R Workspace* what to load, you can import the csv-file:

```
students = read.csv("http://ph.emu.ee/~ktanel/DK_0007/studentsR_eng.csv", header=TRUE,
                    sep=";", dec=",")
```

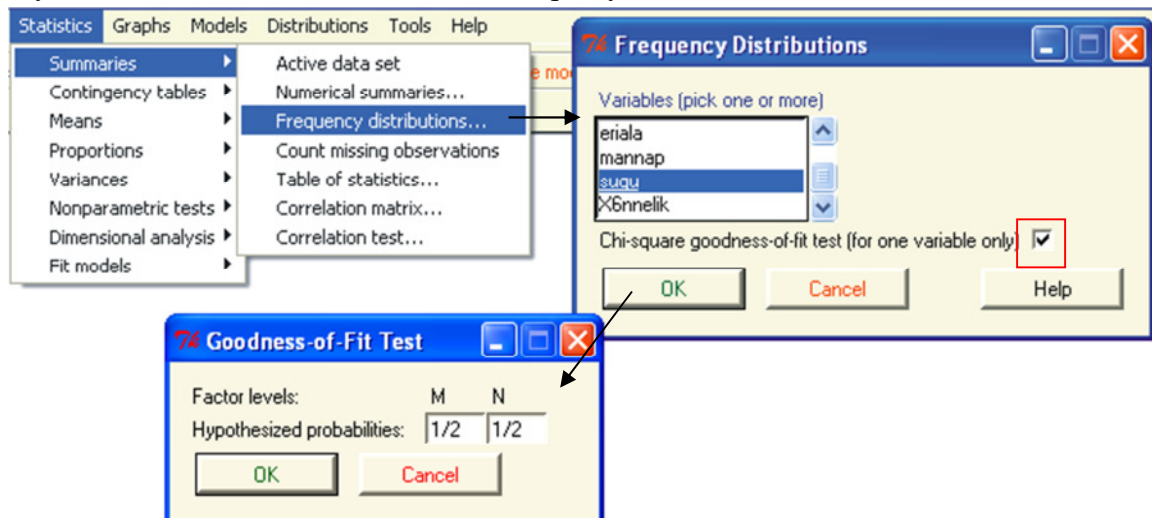
and fix it.

1.4. As an alternative you may also import the *Excel* fail straight into the *R Commander* (*Data -> Import data -> from Excel, Access or dBase data set...*).

2. Frequency tables, χ^2 - and Fisher exact test

2.1. The command *Statistics -> Summaries -> Frequency distributions...* in *R Commander*'s menus allows to construct the frequency tables of nonnumeric traits and additionally to test using the χ^2 -test, are the empirical frequencies counted from data different from some theoretical values.

For example you can test are both sexes distributed equally (50:50) or not:



```

> .Table <- table(students$sugu)

> .Table # counts for sugu

 M N
21 79
Counts of male and female students

> 100*.Table/sum(.Table) # percentages for sugu

 M N
21 79
Relative frequencies (%) of male and female students (as in our dataset the
number of students is 100, then the relative frequencies are equal to counts).

> .Probs <- c(0.5,0.5)

> chisq.test(.Table, p=.Probs)

Chi-squared test for given probabilities

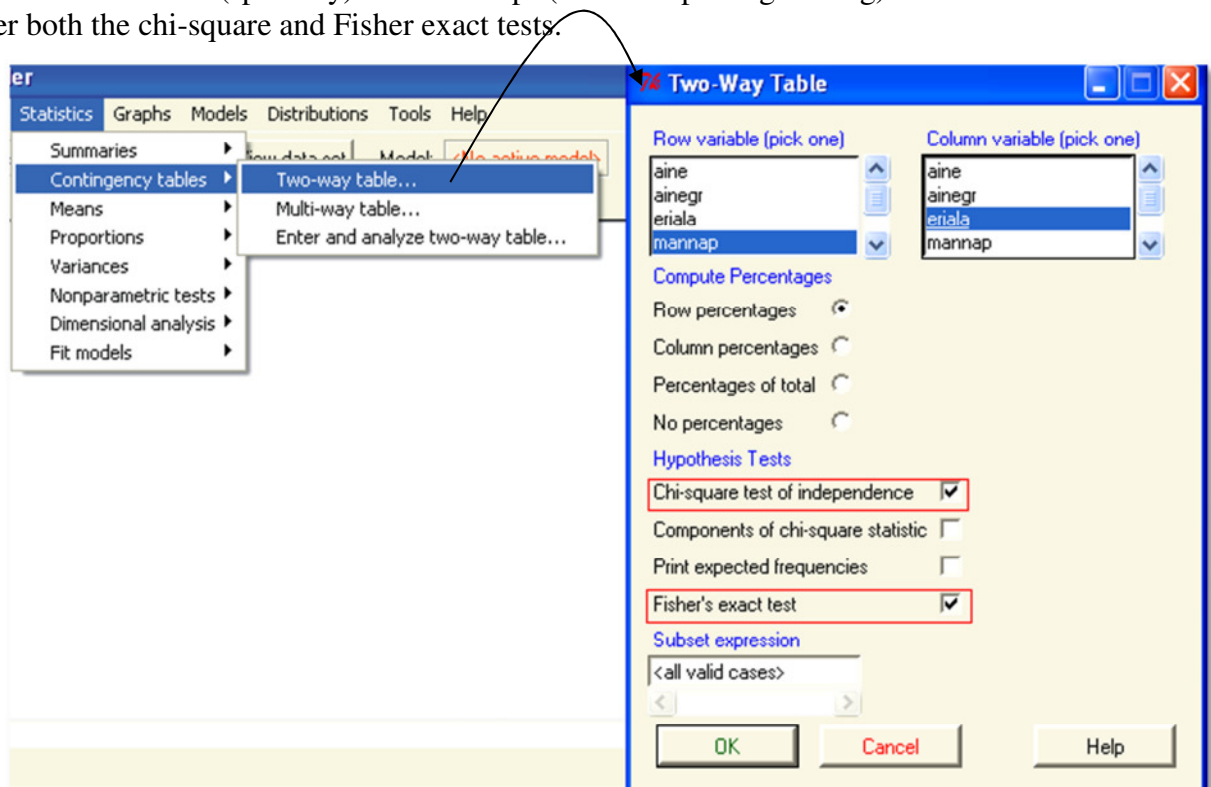
data: .Table
X-squared = 33.64, df = 1, p-value = 6.631e-09
    
```

The results of χ^2 -tests ($p = 6.631 \cdot 10^{-9}$) are saying, that the male and female students are not distributed 50-50.

- Try the same procedure with some other trait or theoretical frequencies.
- To you understand all, what *R Commander* prints into the ‘Script Window’?

2.2. 2-dimentional frequency tables

- Are the traits ‘eriala’ (specialty) and ‘mannap’ (semolina porridge eating) related or not? Order both the chi-square and Fisher exact tests.



```

> .Table <- xtabs(~mannap+eriala, data=students)

> .Table
      eriala
mannap  LAT LKI
Ei      23  23
Jah     22  26
Nii ja naa  2   4

> rowPercents(.Table) # Row Percentages
      eriala
mannap  LAT LKI Total Count
Ei      50.0 50.0   100     46
Jah     45.8 54.2   100     48
Nii ja naa 33.3 66.7   100     6

> .Test <- chisq.test(.Table, correct=FALSE)

> .Test

      Pearson's Chi-squared test

data:  .Table
X-squared = 0.6423, df = 2, p-value = 0.7253

> remove(.Test)

> fisher.test(.Table)

      Fisher's Exact Test for Count Data

data:  .Table
p-value = 0.7681
alternative hypothesis: two.sided

> remove(.Table)

```

The command used by *R Commander*

```
xtabs(~mannap+eriala, data=students)
```

is an alternative to the command

```
table(students$mannap, students$eriala)
```

The result of the command

```
rowPercents(xtabs(~mannap+eriala,
data=students))
```

is almost the same as the result of the command

```
100*prop.table(table(students$mannap,
students$eriala), 1)
```

The second argument '1' of the function `prop.table`

mean the calculation of row percents.

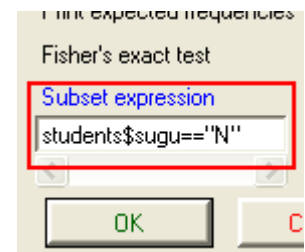
Try the alternative commands.

Both the χ^2 - and Fisher exact test (functions `chisq.test` and `fisher.test`) did not reject the nullhypothesis – the students' speciality and porridge eating are not related ($p = 0,73$ and $p = 0,77$, correspondingly).

- In many analyses the *R Commander* allows to determine the subset of the data to analyze typing the appropriate expression into the box named *<Subset expression>*.

For example modeling the 2-dimentional frequency table you can analyze only women' data by typing into the *<Subset expression>* box the command specifying the rows in dataset 'students' which to use: `students$sugu=="N"`.

As a result *R Commander* performs the analysis and writes into the script window the command



```
xtabs(~eriala+mannap, data=students, subset=students$sugu=="N")
```

Although the commands written by *R Commander* are little bit different from the basic *R* language (did you understand them?)[✧], the following commands will give the same results:

```
table(students$mannap[students$sugu=="N"], students$eriala[students$sugu=="N"])
chisq.test(table(students$mannap[students$sugu=="N"], students$eriala[students$sugu=="N"]))
fisher.test(table(students$mannap[students$sugu=="N"], students$eriala[students$sugu=="N"]))
```

- By the way, did you mentioned the **warning** in *R Commander*'s message window? What this means?

This means that analysing only women is the dataset too small for chi-square test (chi-square test assumes, that expected frequencies in all table cells are at least 5 or bigger).

If the frequencies are less than 5, the teststatistic can not follow the χ^2 -distribution any more and the calculated p-values can be wrong.

```
> .Test$expected # Expected Counts
      mannap
eriala      Ei      Jah Nii ja naa
LAT 19.49367 21.72152      2.78481
LKI 15.50633 17.27848      2.21519
```

What to do?

One variant is to apply the corresponding Monte-Carlo test, which constructs randomly frequency tables with the given row and column sums, calculates in all cases the teststatistic value and estimates in such a way the teststatistic distribution corresponding to the null hypothesis.

This test can be applied in *R* with the function `chisq.test` with the help of additional argument `simulate.p.value=TRUE`, if you wish to change the number of repeatedly constructed frequency tables (default value is 2000), the additional argument `B=5000` should be used (in this case the 5000 frequency tables will be constructed randomly).

As the frequency tables are constructed randomly the test result can be little different at different runs. For example:

```
Pearson's Chi-squared test with simulated p-value (based on 2000 replicates)
data: table(students$mannap[students$sugu == "N"], students$eriala[students$sugu == "N"])
X-squared = 0.816, df = NA, p-value = 0.6747
```

or

```
Pearson's Chi-squared test with simulated p-value (based on 2000 replicates)
data: xtabs(~eriala + mannap, data = students, subset = students$sugu == "N")
X-squared = 0.816, df = NA, p-value = 0.6702
```

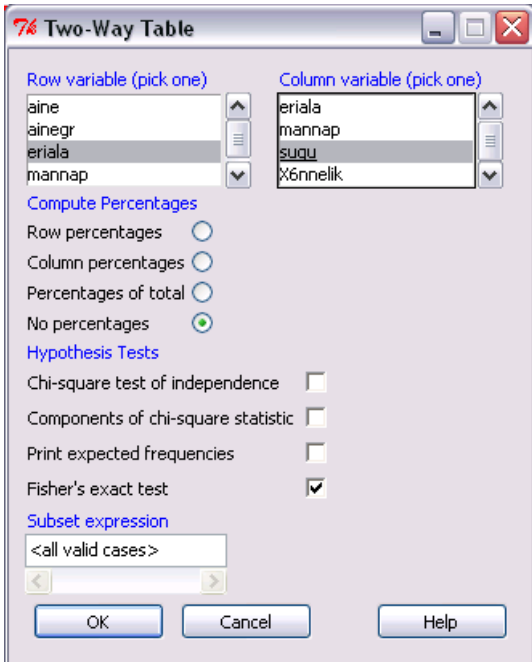
Try, will you also get different results.

As an alternative the Fisher exact test can be used, which constructs all possible frequency tables with given row and column sums and calculates the exact p-value. NB! In case of large datasets and/or frequency tables (with many rows and columns) *R* performs also the Fisher exact test following the Monte-Carlo method ... (otherwise it can take hours or days to get a result).

[✧] It is quite usual for *R Commander*, that it uses the commands not in the simplest form and sometimes the *R Commander* commands did not work in the base *R* – this is the reason of installing the *R Commander* with amount of other packages.

- If the analysed frequency table is 2×2-table, the function `fisher.test` calculates also the odds ratio (OR) and its 95% confidence interval.

For example to test is the men-women ratio among veterinary medicine students and animal science students different (are the sex and speciality related) the Fisher exact test can be performed.



You can order this test from *R Commander* menus

or type and run the corresponding command from the script window:

```
fisher.test(students$seriala, students$sugu)
```

Result:

```

Fisher's Exact Test for Count Data

data: .Table
p-value = 0.001031
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.02360176 0.51641371
sample estimates:
odds ratio
 0.1350761
    
```

Based on the p-value ($p = 0.001$) the alternative hypothesis that the sex and speciality are related is proved (men-women ratio among veterinary medicine students and animal science students is statistically significantly different).

Odds ratio $OR = 0.135$, which is approximately estimable from the frequency tabel as the ratio $(3/18) / (44/35) \approx 0.133$ shows that among veterinary medicine students is the chance to met the male student 0.135 times smaller than among animal science students.

	sugu	
eriala	M	N
LAT	3	44
LKI	18	35

Exactly the same is the odds ratio if we change the table rows and columns:

```
table(students$sugu, students$seriala) ->
```

	LAT	LKI
M	3	18
N	44	35

$OR = (3/44) / (18/35) \approx 0.133$ (*R* gives the estimate 0.135) – among male students is the chace to met the veterinary medicine student 0.135 times smaller than among female students.

The 95% confidence interval of odds ratio is (0,024; 0,516). As this interval does not contain 1 ($OR = 1$ if there is no relationship) the alternative hypothesis that the men-women ratio among veterinary medicine students and animal science students different (sex and speciality related) is proved.

3. Correlation analysis

3.1.

Find the correlations between numerical traits (except the year) using

a) whether the script window in *R Commander*:

```
cor(students[,c("bmi", "kaal", "mat", "peaymb", "pikkus")], use="pair")
```

Remarks:

⌘ the missing first argument (before comma) in

```
[,c("bmi", "kaal", "mat", "peaymb", "pikkus")]
```

says that all rows must be used, the second argument forms the list of used columns;

⌘ simple command `cor(students)` finds correlations between all numerical variables in dataset;

⌘ `cor(students$bmi, students$kaal)` finds the correlation only between mentioned variables;

⌘ options `use="pair"` or `use="complete.obs"` are necessary, if dataset contains missing values:

the second variant (`use="complete.obs"`) asks *R* to **omit all rows with missing values for any of the study variable**,

the first variant (`use="pair"`) **omits only rows with missing values for variables' pair** (for different correlations different number of observations can be used);

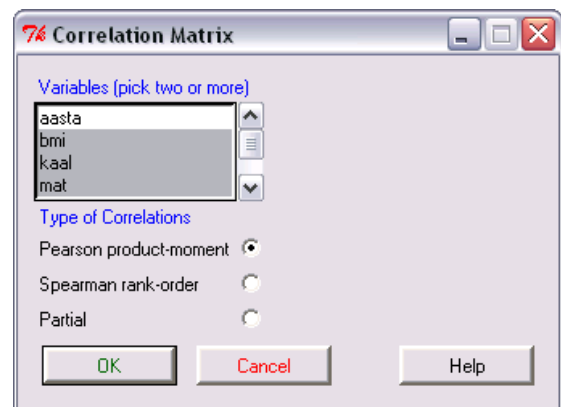
⌘ by default the Pearson correlation coefficients are found,

for Spearman and Kendall correlation coefficients use the options

`method="spearman"` and `method="kendall"`

(also the Pearson correlation coefficient can be specified with `method="pearson"`).

b) or corresponding command from menus *Statistics -> Summaries -> Correlation matrix ...*



3.2. Calculate also the Spearman rank correlation coefficients for the same numerical traits.

Is there any reason to doubt in the linearity of studied relationships?

If there is no big difference between Spearman rank correlation and Pearson linear correlation coefficients, then it is reasonable to use the last (it is more simple and more traditional).

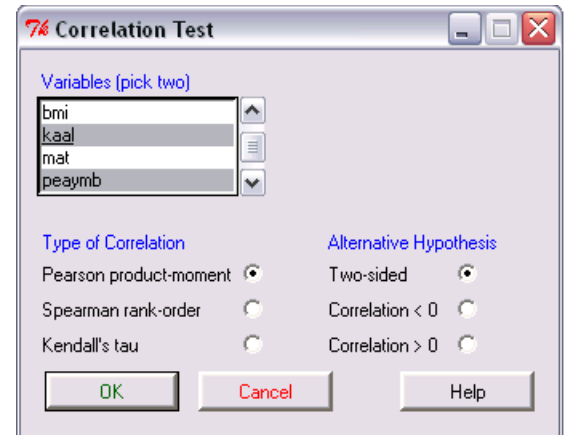
3.3. Are the correlations between head girth ('peaymb) and weight ('kaal'), height ('pikkus') and body mass index ('bmi') statistically significant?

a) You can type the command into the script window using the function `cor.test`:

```
cor.test(students$peaymb, students$kaal, method="pearson")
```

b) or use menus

Statistics -> Summaries -> Correlation test ...



If $p < 0.05$, the corresponding relationship is statistically significant.

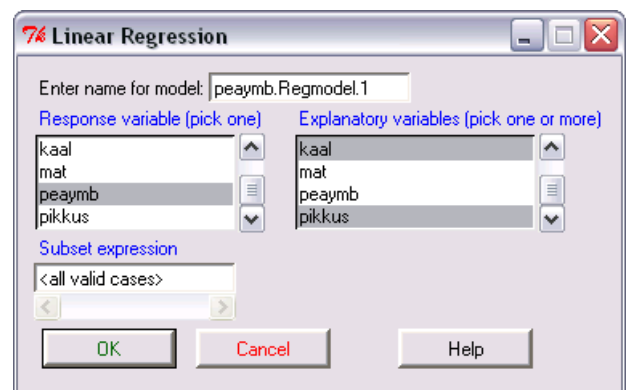
4. Regression analysis

4.1.

- Students usually don't know their head girth ('peaymb'). How well can we predict it based on the weight ('kaal'), height ('pikkus') and body mass index ('bmi') using the linear regression? Are all these arguments necessary in the model?

Statistics -> Fit models -> Linear regression ...

NB! It's important to assign a name to the model, to store and further analyze the model results!!



The corresponding script is

```
peaymb.Regmodel.1 <- lm(peaymb~bmi+kaal+pikkus, data=students)
```

Remarks:

⌘) instead sign $<-$ the sign $=$ can be used;

☞ the equivalent presentation of the model is

```
lm(students$peaymb ~ students$bmi + students$kaal + students$pikkus).
```

The command

```
summary(peaymb.Regmodel.1)
```

is used to print out the model's `peaymb.Regmodel.1` results

Result:

```
Call:
lm(formula = peaymb ~ bmi + kaal + pikkus, data = students)

Residuals:
    Min       1Q   Median       3Q      Max
-9.4769 -1.4241  0.2410  1.8189  9.1708
```

Residuals description

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  76.7956    48.0530   1.598   0.114
bmi          -0.7785     1.0830  -0.719   0.474
kaal          0.4170     0.3690   1.130   0.261
pikkus       -0.1807     0.2817  -0.642   0.523
```

Estimated regression coefficients and corresponding p-values

R-Square, describing the goodness of fit of the model

```
Residual standard error: 2.929 on 88 degrees of freedom
(8 observations deleted due to missingness)
Multiple R-Squared: 0.2836, Adjusted R-squared: 0.2592
F-statistic: 11.61 on 3 and 88 DF, p-value: 1.747e-06
```

Model's statistical significance

- A lot of different model diagnostic parameters and graphs can be calculated from

Models -> Numerical diagnostics and *Models -> Graphs*.

For example *Models -> Numerical diagnostics -> Variance-inflation factors* can be used to measure the multicollinearity of model arguments (this means that arguments are strongly correlated, and as a result the parameters estimates can be misleading).

The multicollinearity is big (and some arguments should omitted from the model), if $VIF > 10$.

```
> vif(peaymb.Regmodel.1)
      bmi      kaal      pikkus
94.4982 174.8210  62.0392
```

So, multicollinearity is too big and at least one argument should be omitted ... Which of them? Look at the correlations – the argument with the biggest correlation with the head girth should usually stay into the model ...

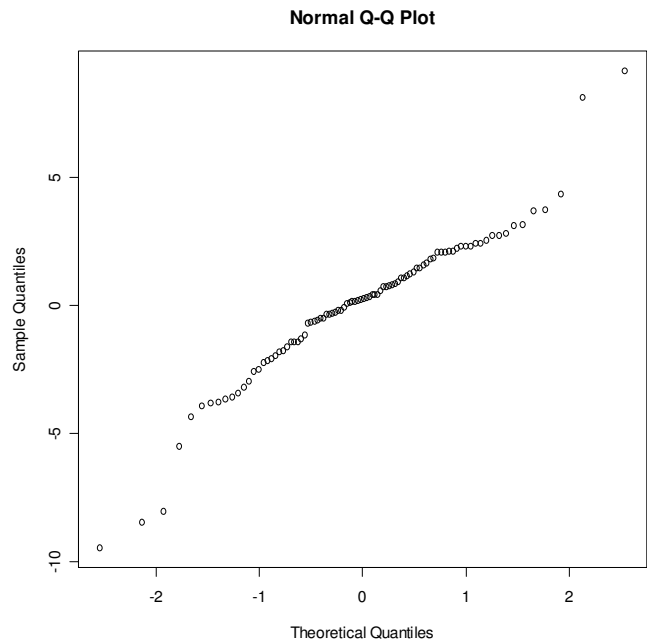
Fit the models with less arguments (giving them own names, for example `peaymb.Regmodel.2` and so on).

- The choice *Models -> Graphs -> Basic diagnostic plots* will give several basic diagnostic plots including the residuals *versus* fitted values plot and residuals normal Q-Q plot (quantile-quantile plot).

For the residuals normal Q-Q plot the also the following command can be used:

```
qqnorm(resid(peaymb.Regmodel.1))
```

The model residuals did not follow exactly the normal distribution (the points did not situate exactly on the diagonal), there are some exceptionally large residuals. If possible, the model should be improved taking into account additional arguments or changing the regression function. But in real life it is quite often impossible to construct better prediction equation and the models with residuals distributed like in figure beside are treated as good enough.



4.2.

- It seems that the optimal model should have only one independent variable – weight ('kaal'). Did you reach to the same conclusion?

The corresponding model output:

```
> peaymb.Regmodel.4 <- lm(peaymb~kaal, data=students)
> summary(peaymb.Regmodel.4)

Call:
lm(formula = peaymb ~ kaal, data = students)

Residuals:
    Min       1Q   Median       3Q      Max
-9.4109 -1.4582  0.2312  1.8826  9.4470

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  45.1738     1.8000   25.10 < 2e-16 ***
kaal          0.1632     0.0277    5.89 6.57e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.907 on 90 degrees of freedom
(8 observations deleted due to missingness)
Multiple R-Squared:  0.2782,    Adjusted R-squared:  0.2702
F-statistic: 34.69 on 1 and 90 DF,  p-value: 6.573e-08
```

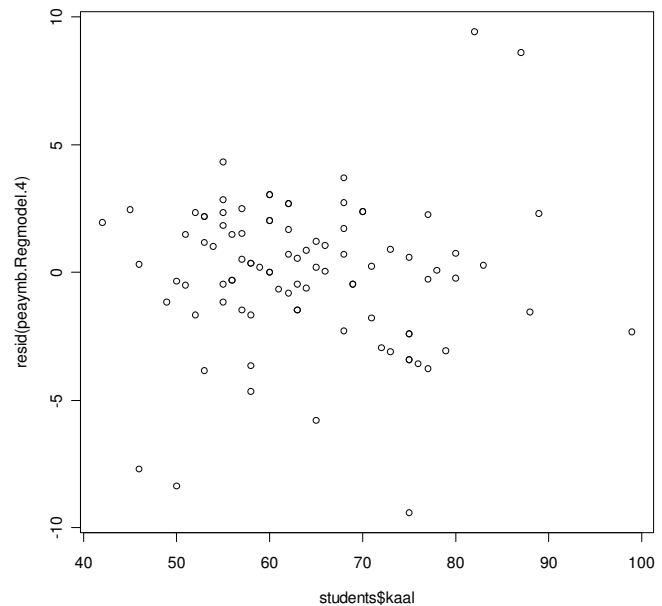
Is this model statistically significant?

What about the regression equation?

- Such a simple model can be illustrated with several 2-dimensional plots.

For example the residuals plot can be ordered using the command.

```
plot(students$kaal, resid(peaymb.Regmodel.4))
```



There is no any clear tendency in the residuals' scatterplot, so the model can be declared suitable.

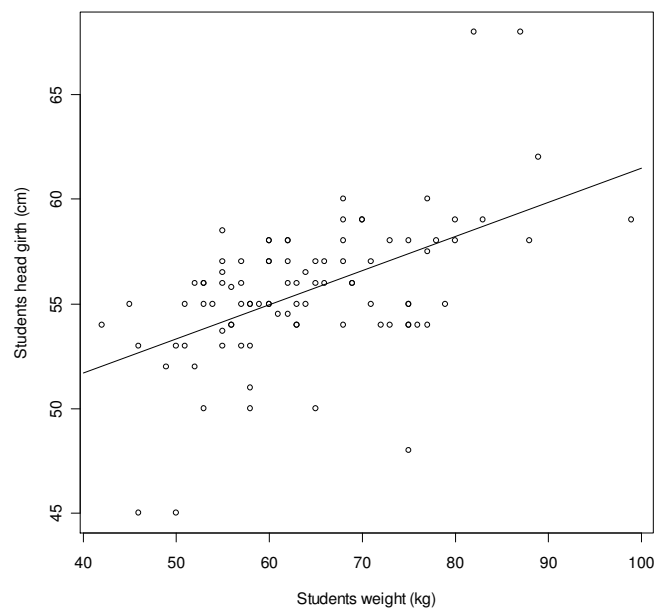
- The predictions in given points (for example for weights 40 and 100 kg) can be calculated by command

```
prediction=predict(peaymb.Regmodel.4, data.frame(kaal=c(40,100)))
```

(the vector of predicted head girths is not printed out but the predicted values are just assigned to the variable `prediction`; if you want to see the predicted values, you should type into the script window the name of the corresponding variable).

The scatter plot with regression line can be drawn running the following commands in script window:

```
plot(students$kaal, students$peaymb,
     xlab="Students weight (kg)",
     ylab="Students head girth (cm)")
lines(c(40,100), prediction)
# draws the line on the plot
```

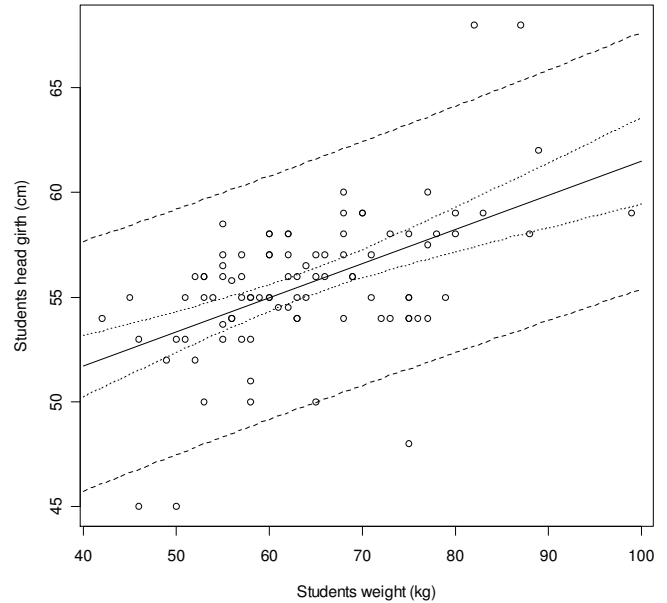


On the same graph the tolerance and confidence intervals can be added using the commands

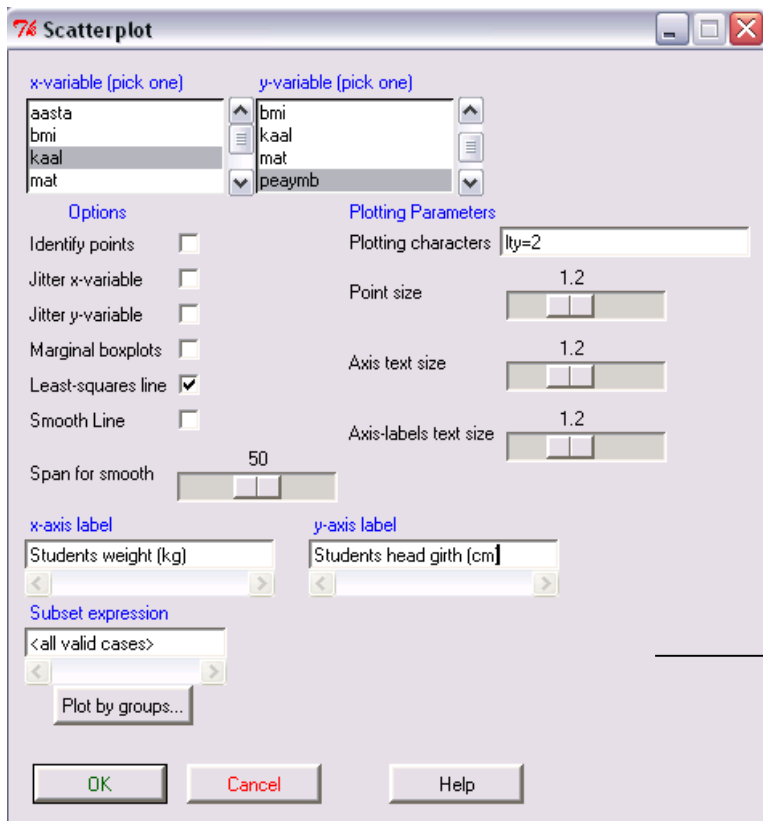
```
prediction=predict(peaymb.Regmodel.4,
  data.frame(kaal=c(40:100)),interval="prediction")
lines(40:100,prediction[,2],lty=2)
lines(40:100,prediction[,3],lty=2)
```

```
prediction=predict(peaymb.Regmodel.4,
  data.frame(kaal=c(40:100)),
  interval="confidence")
lines(40:100,prediction[,2],lty=3)
lines(40:100,prediction[,3],lty=3)
```

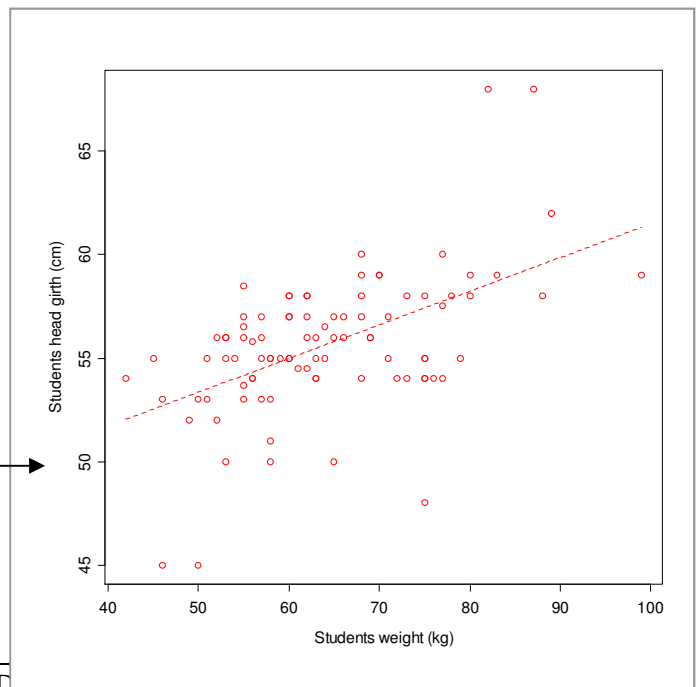
(the 95% tolerance interval shows the area of 95% students' weights and head girths; the 95% confidence interval is the area where the real regression line should be with probability 0.95).



- The similar plot can be drawn also in *R Commander*:
Graphs -> Scatterplot ...



The confidence and tolerance intervals are not selectable in menus but these can be added later using the commands `predict` and `lines` in script window.



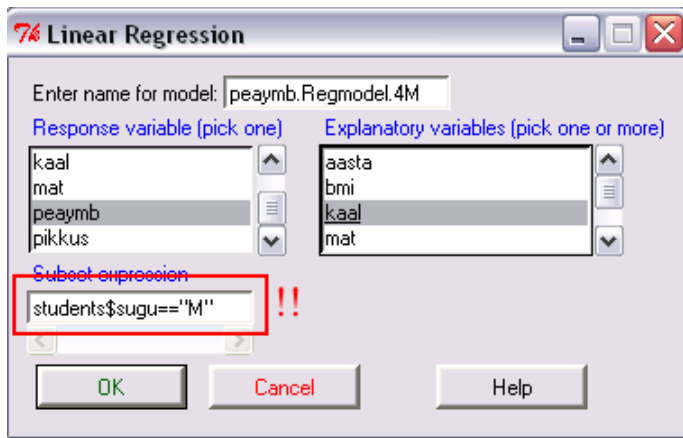
4.3. Is the regression equation similar for both sexes?

- You can build the models separately for men and women running the following commands in script window

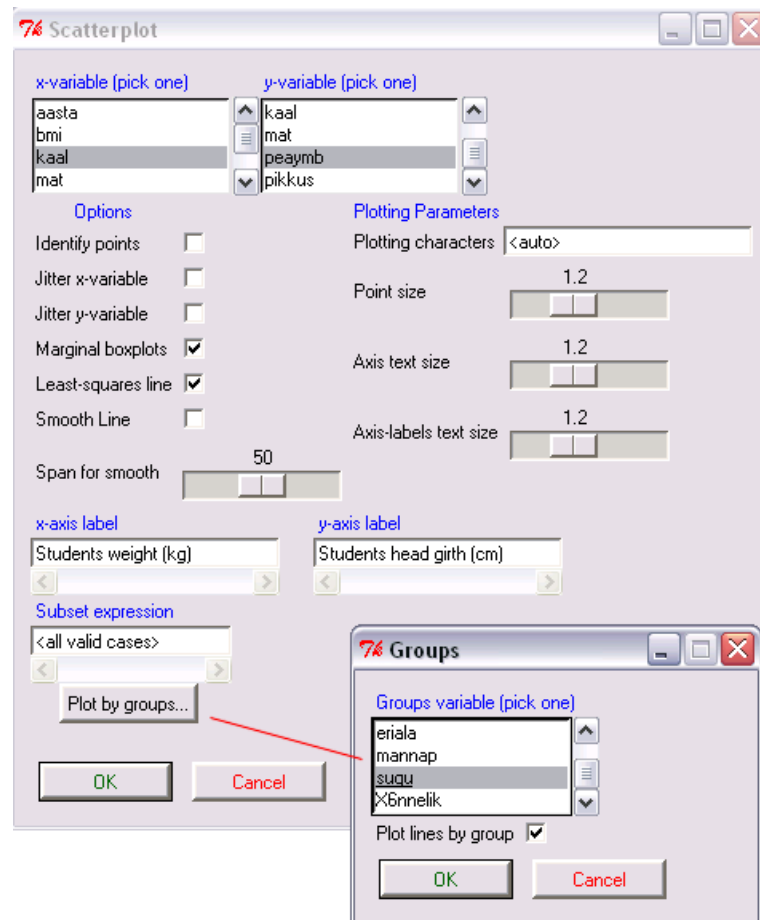
```
peaymb.Regmodel.4M <- lm(peaymb~kaal, data=students[students$sugu=="M",])
summary(peaymb.Regmodel.4M)
```

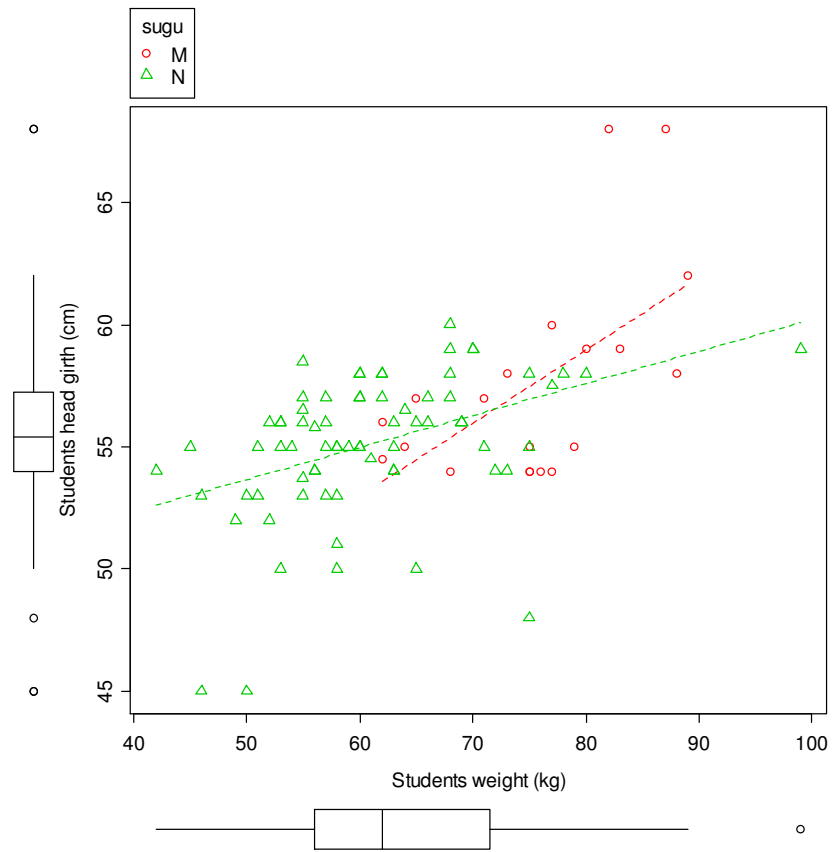
```
peaymb.Regmodel.4N <- lm(peaymb~kaal, data=students[students$sugu=="N",])
summary(peaymb.Regmodel.4N)
```

Or in R Commander: *Statistics -> Fit models -> Linear regression ...*



- The simplest way to draw a scatter plot with different regression lines for both sexes is to use the *Graphs -> Scatterplot ...* in R Commander.





4.4.

And finally try to understand, what kind of graphs can be produced by choosing *Graphs -> Scatterplot matrix...* in R Commander.